A Type-Logical Perspective on Minimalist Derivations

Thomas L. Cornell
University of Tübingen
Seminar für Sprachwissenschaft
cornell@sfs.nphil.uni-tuebingen.de

August 10, 1997

1 Introduction

¶1. Many researchers in the past few years have remarked on the apparent relation between ideas from Chomsky's minimalist program (Chomsky 1995) and fundamental ideas in categorial grammar (e.g., Morrill 1994, Moortgat 1996).

The main feature of minimalist derivations which has suggested a connection with categorial grammar is the treatment of *feature checking* in Chapter 4 of Chomsky (1995), which is very reminiscent of the basic *cancellation rules* of categorial grammar. Consider the following instance of a $L\backslash$ rule, a "rule of use" in a Gentzen style presentation of a Lambek calculus.

$$\frac{K \Rightarrow K \quad V \Rightarrow V}{(K, K\backslash_{\lhd} V)^{\lhd} \Rightarrow V} L\backslash \tag{1}$$

Suppose we interpret K as the case feature of a DP, and the $\lhd$ mode of combination as specifier attachment. Then this instance of a $L\backslash$ rule corresponds fairly closely to the minimalist idea that an expression marked with a case feature can cancel that feature in a checking configuration (e.g., the specifier position) with a head marked with the same feature. In effect, then, the axiom $K \Rightarrow K$ marks the feature as cancelled.

¶2. Chomsky's principle of *Last Resort*, though it appears in several forms in Chomsky (1995), essentially amounts to the following once a theory of feature checking is in place.

> Transformational derivations should be driven by the need to check features.

In a deductive setting this means neither more nor less than that:

> Proofs are driven by the need to reduce a theorem to axioms.

so this key idea of the minimalist program receives a purely formal explanation.

¶3. The features which drive transformational derivations are often *subcategorial*, "inflectional" features like Case or Wh. So lexical items can be assigned to multiple types, and furthermore, to assure feature-driven derivations, they must be assigned to multiple types in a resource sensitive manner. So we are led to conclude that,

> Subcategorial features can be resources, too.

at least insofar as they drive derivations.

Accordingly, we allow mulitplicative product types in our lexicon. For example, a Wh determiner like *which*, needing Case and selecting a noun type complement might be represented with a formula like

$$(D \bullet K \bullet W)/N,$$

where we use K for Case and W for Wh. (We treat $\bullet$ as right branching by default, so as to avoid unnecessary parentheses.)

Just as the combinatorial potential of phrases is given in advance by the type of the lexical head, so in this system the movement potential of a phrase is given in advance by the type of its head. For example, the WhP headed by *which*, given this type assignment, will have to occupy three possibly distinct syntactic positions in order to cancel its three "positive" features.

> Roughly: '/' and '\' say how to build phrases, '$\bullet$' says how to build chains.

2 Numerations and Sequents

¶4. As presented in Chomsky (1995), a derivation proceeds from an initially given *numeration*, which is supposed to be a *multiset* of lexical item occurrences.

The numeration is then subjected to the action of a pair of *generalized transformations*, **Merge** and **Move**, which transform the raw lexical resources of the numeration into an LF representation, delivering an intermediate representation for PF spell-out along the way.

¶5. In the setting of a grammar logic we want the lexical item instances to be type formulas, and we want to isolate in particular collections of such lexical formulas which can be combined to form sentences. We represent the fundamental decision problem of a grammar logic as the question of whether or not a sequent of the form

$$\text{lexical resources} \Rightarrow \text{sentence-category}$$

is derivable.

Grafting minimalist ideas into this setting seems fairly natural: we are interested in sequents of the form

$$\text{Numeration} \Rightarrow C$$

where C (complementizer) is the sentence category.

3 Checking Configurations

¶6. Chomsky (1995) defines the *checking domain* of a head in such a way that it includes all head-internal positions and specifier positions, and it excludes the complement position, which becomes the *attraction domain*. The flow of movement is then from the attraction domain and into the checking domain.

¶7. Chomsky (1995) lacks a theory of selection. To a point one can imagine categorial grammar as providing a theory of selection and the minimalist program as providing a theory of attraction. We wish to combine both theories here, to obtain a theory which is more *uniformly resource conscious*. So here even the complement position will be a checking position.

¶8. Note that in the case of the Spec-Head configuration, and even—following head movement—in some cases of the Complement-Head configuration as well, clashing pairs of features are not "in configuration" with each other. That is, the standard $L/$ and $L\backslash$ rules of the Lambek calculus require that the functor and its argument be sisters in the sequent antecedent. Moortgat and Oehrle (1997) offer two sorts of solutions for this sort of problem: "structural normalization" and "structural transparency". The former adds controlled structural rules so as to allow the relevant resources to be placed into the canonical sister configuration; the latter allows intervening structure to be ignored in the relevant cases. Here we use the former approach: once the generalized transformation **Move** has acted to place features in checking configurations, a further suite of structural rules allows the feature to be cancelled to move into position as a sister of the cancelling functor.

4 Connectives of the Grammar Logic

$\otimes$ The numeration forming operator.

We will use a "sugared notation" for the corresponding structures. That is, we use unbracketed commas to actually present the numeration part of a sequent, writing $A, B \Rightarrow C$, rather than $(A, B)^{\otimes} \Rightarrow C$. For example, the antecedent of the sequent $A, B, (C, D)^{>} \Rightarrow E$ is a numeration with 3 elements: two formulas and the tree $(C, D)^{>}$. (See below for $>$.) Unbracketed substructures are granted free access to permutation and association rules.

$\bullet$ Lexical product. Alias presyntactic chain formation operator. Also the "normalized checking configuration" forming comma.

$\backslash, /$ Its residuals.

$\Diamond_0$ "Micromovement" licenser. For structural normalization of checking configurations.

$\Box_0^{\downarrow}$ Its residual. All formal features are marked with this.

$\Diamond_1$ "Macromovement" licenser. For implementing **Move**. All chains are marked with this.

So the type of *which* given above should be emended to read:

$$(\Diamond_1(\Box_0^{\downarrow} D \bullet \Box_0^{\downarrow} K \bullet \Box_0^{\downarrow} W))/N.$$

- $>$ Complement attachment product.
- $\lhd$ Specifier attachment product.
- $\blacktriangleleft$ Prefix attachment product.
- $\blacktriangleright$ Suffix attachment product.

The notation for these last four "tree-building" products is derived from the algebraic systems for minimalist derivations of Stabler (Stabler forthcoming, 1997, 1996). They can be thought of as *headed* operators, with the head greater than the nonhead.

In the rules in (5), '$\star$' should be taken as ranging over all the above binary products: $\{\otimes, \bullet, >, \lhd, \blacktriangleleft, \blacktriangleright\}$.

$$A \Rightarrow A \quad Axiom \qquad \frac{\Gamma \Rightarrow A \quad \Delta[A] \Rightarrow B}{\Delta[\Gamma] \Rightarrow B}\,Cut \tag{2}$$

$$\frac{\Gamma \Rightarrow A \quad \Delta[B] \Rightarrow C}{\Delta[(B/A, \Gamma)^{\bullet}] \Rightarrow C}\,L/ \qquad \frac{(\Delta, A)^{\bullet} \Rightarrow B}{\Delta \Rightarrow B/A}\,R/ \tag{3}$$

$$\frac{\Gamma \Rightarrow A \quad \Delta[B] \Rightarrow C}{\Delta[(\Gamma, A\backslash B)^{\bullet}] \Rightarrow C}\,L\backslash \qquad \frac{(A, \Delta)^{\bullet} \Rightarrow B}{\Delta \Rightarrow A\backslash B}\,R\backslash \tag{4}$$

$$\frac{\Delta[(A, B)^{\star}] \Rightarrow C}{\Delta[A \star B] \Rightarrow C}\,L\star \qquad \frac{\Delta_1 \Rightarrow A \quad \Delta_2 \Rightarrow B}{(\Delta_1, \Delta_2)^{\star} \Rightarrow A \star B}\,R\star \tag{5}$$

$$\frac{\Gamma[A] \Rightarrow B}{\Gamma[(\Box^{\downarrow} A)^{\Diamond}] \Rightarrow B}\,L\Box^{\downarrow} \qquad \frac{(\Gamma)^{\Diamond} \Rightarrow A}{\Gamma \Rightarrow \Box^{\downarrow} A}\,R\Box^{\downarrow} \tag{6}$$

$$\frac{\Gamma[(A)^{\Diamond}] \Rightarrow B}{\Gamma[\Diamond A] \Rightarrow B}\,L\Diamond \qquad \frac{\Gamma \Rightarrow A}{(\Gamma)^{\Diamond} \Rightarrow \Diamond A}\,R\Diamond \tag{7}$$

5 Packages of Structural Rules

5.1 Merge Package

$$\frac{\Gamma, (\Delta_1, \Delta_2)^{>} \Rightarrow A}{\Gamma, \Delta_1, \Delta_2 \Rightarrow A}\,Mg1 \tag{8}$$

$$\frac{\Gamma, (\Delta_1, \Delta_2)^{\lhd} \Rightarrow A}{\Gamma, \Delta_1, \Delta_2 \Rightarrow A}\,Mg2 \tag{9}$$

Rule $Mg1$ applies only if Δ_1 is a functor formula. Otherwise rule $Mg2$ applies.

We require in addition that the Δ_i not contain any instances of $\otimes$ or its corresponding structure-building comma.

5.2 Move (XP) Package

$$\frac{\Gamma[((\Delta_1)^{\diamond_1},(\Delta_2)^{\diamond_1})]\Rightarrow A}{\Gamma[((\Delta_1,\Delta_2))^{\diamond_1}]\Rightarrow A}\,K \tag{10}$$

$$\frac{\Gamma[((\Delta_2)^{\diamond_1},\Delta_1)^{\lhd}]\Rightarrow A}{\Gamma[(\Delta_1,(\Delta_2)^{\diamond_1})^{>}]\Rightarrow A}\,Mv1 \tag{11}$$

$$\frac{\Gamma[((\Delta_2)^{\diamond_1},(\Delta_1,\Delta_3)^{>})^{\lhd}]\Rightarrow A}{\Gamma[(\Delta_1,((\Delta_2)^{\diamond_1},\Delta_3)^{\lhd})^{>}]\Rightarrow A}\,Mv2 \tag{12}$$

$$\frac{\Gamma[((\Delta_2)^{\diamond_1},(\Delta_1,\Delta_3)^{\lhd})^{\lhd}]\Rightarrow A}{\Gamma[(\Delta_1,((\Delta_2)^{\diamond_1},\Delta_3)^{\lhd})^{\lhd}]\Rightarrow A}\,Mv3 \tag{13}$$

Our system always restricts inference rules to act on very local configurations of resources. As a result this system looks rather different in detail than a traditional transformational approach. For purposes of comparison, it is best not to think of the application of individual Mv rules as each implementing a movement step. Rather, an application of the generalized transformation **Move** corresponds to the entire subproof between the application of the first Mv rule to an expression to the application of the Mm rule, which switches the system from **Move** mode to feature cancellation mode.

¶9. Locality conditions may be placed as side conditions on these rules. For example, (reading bottom up) a formula cannot cross another (c-commanding) formula with the same feature (with the same polarity). Nor can it move away from a head with an oppositely charged version of the same feature. Take for example $Mv2$. If $\Delta_2 = A$ and $\Delta_3 = A\backslash B$, for example, then the application of $Mv2$ should be blocked. In $Mv3$, if (the head of) Δ_3 contains an oppositely charged version of a feature found in Δ_2, or Δ_1 contains a similarly charged feature, then the application of the rule is blocked. In all these cases we would be violating Shortest Move, alias the Minimal Link Condition.

¶10. Also so far nothing has required that movement extend the structure, i.e., that all movement should be to the root. There is no way to do this with local movement, since quite evidently individual "sub-moves" cannot be to the root in general. Possibly we could try to restrict the formation of normalized checking domains in the checking component to force the head term to be the head of the entire tree. More generally, since the switch from $\diamond_1$ to $\diamond_0$ marks the end of a movement, such conditions should probably be placed on the feature checking package rather than on the **Move** package.

5.3 Feature Checking Package ("Micro-Movement")

$$\frac{\Gamma[(\Delta)^{\diamond_0}] \Rightarrow A}{\Gamma[(\Delta)^{\diamond_1}] \Rightarrow A}\,Mm \tag{14}$$

$$\frac{\Gamma[(\Delta_1, (\Delta_2)^{\diamond_0})^{\bullet}] \Rightarrow A}{\Gamma[(\Delta_1, (\Delta_2)^{\diamond_0})^{>}] \Rightarrow A}\,F0c \qquad \frac{\Gamma[((\Delta_1)^{\diamond_0}, \Delta_2)^{\bullet}] \Rightarrow A}{\Gamma[((\Delta_1)^{\diamond_0}, \Delta_2)^{\lhd}] \Rightarrow A}\,F0s \tag{15}$$

$$\frac{\Gamma[((\Delta_1, (\Delta_2)^{\diamond_0})^{\bullet}, \Delta_3)^{>}] \Rightarrow A}{\Gamma[(\Delta_1, ((\Delta_2)^{\diamond_0}, \Delta_3)^{\bullet})^{>}] \Rightarrow A}\,F1c \qquad \frac{\Gamma[(\Delta_2, ((\Delta_1)^{\diamond_0}, \Delta_3)^{\lhd})^{\lhd}] \Rightarrow A}{\Gamma[(((\Delta_1)^{\diamond_0}, \Delta_2)^{\bullet}, \Delta_3)^{\lhd}] \Rightarrow A}\,F1s \tag{16}$$

$$\frac{\Gamma[(\Delta_2, ((\Delta_1)^{\diamond_0}, \Delta_3)^{\lhd})^{\lhd}] \Rightarrow A}{\Gamma[((\Delta_1)^{\diamond_0}, (\Delta_2, \Delta_3)^{\lhd})^{\lhd}] \Rightarrow A}\,F2 \tag{17}$$

$$\frac{\Gamma[(((\Delta_1)^{\diamond_0}, \Delta_2)^{\bullet}, \Delta_3)^{>}] \Rightarrow A}{\Gamma[((\Delta_1)^{\diamond_0}, (\Delta_2, \Delta_3)^{>})^{\lhd}] \Rightarrow A}\,F3 \tag{18}$$

These rules act to break up a lexical product in one place, and form a new one in another. The c variants act to remove a feature from complement position, and the s variants act in the same way on specifiers. Note however that there can be multiple specifiers in Chomsky's system, but only a single complement. So the rules which (read bottom up) move a feature from the specifier to the head are somewhat more complex than those moving a feature from the complement to the head.

6 (Minimalist) Derivation as Proof Search

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{D \Rightarrow D \quad V \Rightarrow V}{(V/D, D)^{\bullet} \Rightarrow V}\,L/}{(V/D, (\Box_0^{\downarrow} D)^{\diamond_0})^{\bullet} \Rightarrow V}\,L\Box_0^{\downarrow}}{(V/D, (\Box_0^{\downarrow} D)^{\diamond_0})^{>} \Rightarrow V}\,F0c}{(V/D, (\Box_0^{\downarrow} D)^{\diamond_1})^{>} \Rightarrow V}\,Mm}{V/D, (\Box_0^{\downarrow} D)^{\diamond_1} \Rightarrow V}\,Mg}{V/D, \diamond_1 \Box_0^{\downarrow} D \Rightarrow V}\,L\diamond_1$$

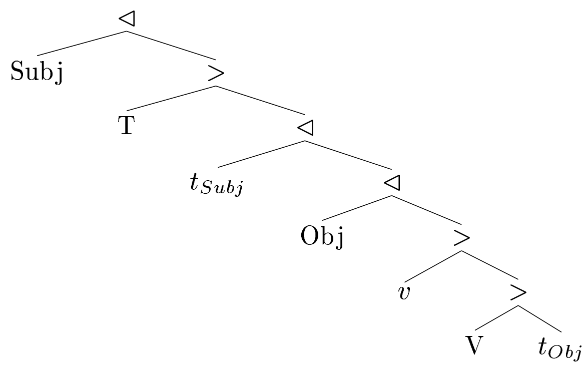
Figure 1: An elementary derivation, cancelling one selection feature.

$$
\dfrac{\dfrac{D \Rightarrow D \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{K \Rightarrow K \quad V \Rightarrow V}{(K, K\backslash V)^{\bullet} \Rightarrow V}\,L\backslash}{((\Box_0^{\downarrow}K)^{\Diamond_0}, K\backslash V)^{\bullet} \Rightarrow V}\,L\Box_0^{\downarrow}}{((\Box_0^{\downarrow}K)^{\Diamond_0}, K\backslash V)^{\lhd} \Rightarrow V}\,F0s}{((\Box_0^{\downarrow}K)^{\Diamond_1}, K\backslash V)^{\lhd} \Rightarrow V}\,Mm}{(K\backslash V, (\Box_0^{\downarrow}K)^{\Diamond_1})^{>} \Rightarrow V}\,Mv1}{(((K\backslash V)/D, D)^{\bullet}, (\Box_0^{\downarrow}K)^{\Diamond_1})^{>} \Rightarrow V}\,L/}{\dfrac{(((K\backslash V)/D, (\Box_0^{\downarrow}D)^{\Diamond_0})^{\bullet}, (\Box_0^{\downarrow}K)^{\Diamond_1})^{>} \Rightarrow V}{\dfrac{((K\backslash V)/D, ((\Box_0^{\downarrow}D)^{\Diamond_0}, (\Box_0^{\downarrow}K)^{\Diamond_1})^{\bullet})^{>} \Rightarrow V}{\dfrac{((K\backslash V)/D, ((\Box_0^{\downarrow}D)^{\Diamond_1}, (\Box_0^{\downarrow}K)^{\Diamond_1})^{\bullet})^{>} \Rightarrow V}{\dfrac{(K\backslash V)/D, ((\Box_0^{\downarrow}D)^{\Diamond_1}, (\Box_0^{\downarrow}K)^{\Diamond_1})^{\bullet} \Rightarrow V}{\dfrac{(K\backslash V)/D, ((\Box_0^{\downarrow}D, \Box_0^{\downarrow}K)^{\bullet})^{\Diamond_1} \Rightarrow V}{\dfrac{(K\backslash V)/D, ((\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K))^{\Diamond_1} \Rightarrow V}{(K\backslash V)/D, \Diamond_1(\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K) \Rightarrow V}\,L\Diamond_1}\,L\bullet}\,K}\,Mg}\,Mm}\,F1c}\,L\Box_0^{\downarrow}
$$

Figure 2: A slightly more complex derivation, canceling one selection feature and one formal feature.

7 A More Elaborate Derivation

In Chomsky (1995), the following structure is considered for ordinary transitive clauses (disregarding head movement, and assuming overt object raising), along with a variant in which the overt object outscopes the subject trace.



Assuming that accusative case is assigned by the "light verb" v, we need something like the following lexical assignments.

$$v \mapsto (K\backslash(D\backslash(\Diamond_1\Box_0^{\downarrow}V)))/V \tag{19}$$

$$V \mapsto (\Diamond_1\Box_0^{\downarrow}V)/D \tag{20}$$

The following is a theorem in our system.

$$\begin{split}(K\backslash T)/V, (K\backslash(D\backslash(\Diamond_1\Box_0^{\downarrow}V)))/V, \Diamond_1(\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K)_{Subj},\\ (\Diamond_1\Box_0^{\downarrow}V)/D, \Diamond_1(\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K)_{Obj} \Rightarrow T\end{split} \tag{21}$$

Observe that the first feature to be cancelled must be the D feature of one of the two DPs. Choosing the one marked "Obj", that leaves us with the following goal sequent.

$$\begin{split}(K\backslash T)/V, (K\backslash(D\backslash(\Diamond_1\Box_0^{\downarrow}V)))/V, \Diamond_1(\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K)_{Subj},\\ ((\Box_0^{\downarrow}V)^{\Diamond_1}, (\Box_0^{\downarrow}K)^{\Diamond_1})^{>} \Rightarrow T\end{split} \tag{22}$$

Note that we could at this point merge the subject formula with the tree we have built so far. That would put it closer to the light verb's K feature, blocking movement of the object's K feature. But the subject cannot cancel the the verb's K feature itself: it has to discharge its D feature in order to "release" the K feature. So that move is blocked by the nonassociative and noncommutative nature of the $\bullet$ product. Instead we merge the current tree with the light verb.

$$\begin{split}(K\backslash T)/V, \Diamond_1(\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K)_{Subj},\\ ((K\backslash(D\backslash(\Diamond_1\Box_0^{\downarrow}V)))/V, ((\Box_0^{\downarrow}V)^{\Diamond_1}, (\Box_0^{\downarrow}K)^{\Diamond_1})^{>})^{>} \Rightarrow T\end{split} \tag{23}$$

Now we can move the object K feature out of the lower VP.

$$\begin{split}(K\backslash T)/V, \Diamond_1(\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K)_{Subj},\\ ((\Box_0^{\downarrow}K)^{\Diamond_1}, ((K\backslash(D\backslash(\Diamond_1\Box_0^{\downarrow}V)))/V, (\Box_0^{\downarrow}V)^{\Diamond_1})^{>})^{\lhd} \Rightarrow T\end{split} \tag{24}$$

Now we can cancel the V feature. Note that, without a Head Movement package, we could not do this until we had evacuated all uncancelled features from around the head. This trick works, but note that it runs counter to Chomsky's intentions, according to which the head should move first. The rest of the derivation offers nothing new; relevant snapshots of the main goal sequent are given below.

$$(K\backslash T)/V, \Diamond_1(\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K)_{Subj}, ((\Box_0^{\downarrow}K)^{\Diamond_1}, K\backslash(D\backslash(\Diamond_1\Box_0^{\downarrow}V)))^{\lhd} \Rightarrow T \tag{25}$$

$$(K\backslash T)/V, (\Diamond_1(\Box_0^{\downarrow}D \bullet \Box_0^{\downarrow}K)_{Subj}, D\backslash(\Diamond_1\Box_0^{\downarrow}V))^{\lhd} \Rightarrow T \tag{26}$$

$$(K\backslash T)/V, ((\Box_0^{\downarrow}K)^{\Diamond_1}, (\Box_0^{\downarrow}V)^{\Diamond_1})^{\lhd} \Rightarrow T \tag{27}$$

$$((\Box_0^{\downarrow}K)^{\Diamond_1}, ((K\backslash T)/V, (\Box_0^{\downarrow}V)^{\Diamond_1})^{>})^{\lhd} \Rightarrow T \tag{28}$$

Step (25) is the result of cancelling the lower V feature. Step (26) is the resultof cancelling the object's K feature against the higher verb. Then the subject's D feature is cancelled against the verb leading to (27). Finally, after some applications of rules from the **Merge** package, the subject's K feature can move to specifier of T, leaving the verb free to cancel its V feature against the V feature of T, which will then allow the cancellation of the subject's K feature, finishing the proof.

8 Derivation and Representation

$$A : x \Rightarrow A : x \quad Axiom \qquad \frac{\Gamma \Rightarrow \genfrac{}{}{0pt}{}{A}{t} \quad \Delta[\genfrac{}{}{0pt}{}{A}{x}] \Rightarrow \genfrac{}{}{0pt}{}{B}{u}}{\Delta[\Gamma] \Rightarrow \genfrac{}{}{0pt}{}{B}{u[t/x]}}\,Cut \tag{29}$$

$$\frac{\Gamma \Rightarrow \genfrac{}{}{0pt}{}{A}{t} \quad \Delta[\genfrac{}{}{0pt}{}{B}{x}] \Rightarrow \genfrac{}{}{0pt}{}{C}{u}}{\Delta[(\genfrac{}{}{0pt}{}{B/A}{y}, \Gamma)^{\bullet}] \Rightarrow \genfrac{}{}{0pt}{}{C}{u[y(t)/x]}}\,L/ \qquad \frac{(\Delta, \genfrac{}{}{0pt}{}{A}{x})^{\bullet} \Rightarrow \genfrac{}{}{0pt}{}{B}{u}}{\Delta \Rightarrow \genfrac{}{}{0pt}{}{B/A}{\lambda x.u}}\,R/ \tag{30}$$

$$\frac{\Gamma \Rightarrow \genfrac{}{}{0pt}{}{A}{t} \quad \Delta[\genfrac{}{}{0pt}{}{B}{x}] \Rightarrow \genfrac{}{}{0pt}{}{C}{u}}{\Delta[(\Gamma, \genfrac{}{}{0pt}{}{A\backslash B}{y})^{\bullet}] \Rightarrow \genfrac{}{}{0pt}{}{C}{u[y(t)/x]}}\,L\backslash \qquad \frac{(\genfrac{}{}{0pt}{}{A}{x}, \Delta)^{\bullet} \Rightarrow \genfrac{}{}{0pt}{}{B}{u}}{\Delta \Rightarrow \genfrac{}{}{0pt}{}{A\backslash B}{\lambda x.u}}\,R\backslash \tag{31}$$

$$\frac{\Delta[(\genfrac{}{}{0pt}{}{A}{x_1}, \genfrac{}{}{0pt}{}{B}{x_2})^{\star}] \Rightarrow \genfrac{}{}{0pt}{}{C}{u}}{\Delta[\genfrac{}{}{0pt}{}{A\star B}{z}] \Rightarrow \genfrac{}{}{0pt}{}{C}{u[\pi_1 z/x_1, \pi_2 z/x_2]}}\,L\star \qquad \frac{\Delta_1 \Rightarrow \genfrac{}{}{0pt}{}{A}{x_1} \quad \Delta_2 \Rightarrow \genfrac{}{}{0pt}{}{B}{x_2}}{(\Delta_1, \Delta_2)^{\star} \Rightarrow \genfrac{}{}{0pt}{}{A\star B}{\langle x_1, x_2\rangle}}\,R\star \tag{32}$$

$$\frac{\Gamma[\genfrac{}{}{0pt}{}{A}{x}] \Rightarrow \genfrac{}{}{0pt}{}{B}{u}}{\Gamma[\genfrac{}{}{0pt}{}{(\Box^{\downarrow}A)^{\Diamond}}{y}] \Rightarrow \genfrac{}{}{0pt}{}{B}{u[{}^{\vee}y/x]}}\,L\Box^{\downarrow} \qquad \frac{(\Gamma)^{\Diamond} \Rightarrow \genfrac{}{}{0pt}{}{A}{t}}{\Gamma \Rightarrow \genfrac{}{}{0pt}{}{\Box^{\downarrow}A}{{}^{\wedge}t}}\,R\Box^{\downarrow} \tag{33}$$

$$\frac{\Gamma[(\genfrac{}{}{0pt}{}{A}{x})^{\Diamond}] \Rightarrow \genfrac{}{}{0pt}{}{B}{u}}{\Gamma[\genfrac{}{}{0pt}{}{\Diamond A}{y}] \Rightarrow \genfrac{}{}{0pt}{}{B}{u[{}^{\cup}y/x]}}\,L\Diamond \qquad \frac{\Gamma \Rightarrow \genfrac{}{}{0pt}{}{A}{t}}{(\Gamma)^{\Diamond} \Rightarrow \genfrac{}{}{0pt}{}{\Diamond A}{{}^{\cap}t}}\,R\Diamond \tag{34}$$

¶11. We have already observed that lexical type formulas represent "in miniature" the phrases and chains into which the syntax will extend them. In general, lexical types represent functions from phrases to chains. Consider again the assignment to the Wh determiner *which*, ignoring the control modals for the moment:

$$\textit{which} \mapsto (D \bullet K \bullet W)/N \tag{35}$$

One possible term assignment would be something like the following.

$$(D \bullet K \bullet W)/N : \lambda x.\langle(\alpha > x), (\alpha > x), (\alpha > x)\rangle \tag{36}$$

This implements the Copy Theory of Movement. Cancelling the N feature with the $L/$ rule will have the effect of applying this function to the value of the NP, making 3 copies of it. Then as each of the DP's own features are cancelled, one copy will be peeled off of the tuple and attached into the phrase of the cancelling head.

¶12. It is interesting that, as far as proof terms are concerned, it is the $L/$ and $L\backslash$ rules which implement **Merge**. This connection between **Merge** and feature cancellation, or something very much like it, has already been proposed in the literature (Collins 1997, Bobaljik 1995). In particular, it has been proposed that **Move** should be treated as removing uncancelled features from the tree and returning them to the numeration, whence they are later merged again at a higher point. So it will always be **Merge** which (a) builds structure and (b) in so doing places features in checking configurations.

¶13. It is also interesting that the essentials of the chain are formed before any actual syntax has been done. This too is an idea that can already be found in the literature, in the theory of presyntactic chain formation developed in recent years by Michael Brody (e.g., Brody 1995).

This is especially interesting inasmuch as Brody espouses a representational, chain-based perspective on extraction while Collins, Bobaljik and company embrace a derivational, movement-based perspective. Normally, these two perspectives are taken to be competing theories, but in the light of a deductive approach they appear as two sides of the same coin.

¶14. A more traditional view of movement would be obtained with an assignment like the following.

$$(D \bullet K \bullet W)/N : \lambda x.\langle t_\alpha, t'_\alpha, (\alpha > x)\rangle \tag{37}$$

In this version we have committed to spelling out the DP in its highest position.

Intermediate positions are possible. One possibility is suggested by Collins (1997): starting with a copying operation, we then apply partial deletion to the copies ("up to recoverability"). In effect, we just have to be sure that, knitting all the copies back together again, we obtain the features of the original lexical item.

¶15. Now consider the full type assignment for *which*,

$$\textit{which} \mapsto (\Diamond_1(\Box_0^{\downarrow} D \bullet \Box_0^{\downarrow} K \bullet \Box_0^{\downarrow} W)/N \tag{38}$$

and the term assignments in the rules for the unary connectives given above. The symbols '${}^{\cap}\cdot$' and '${}^{\wedge}\cdot$' are meant to represent constructor functions for $\Diamond$ and $\Box^{\downarrow}$, respectively, with '${}^{\cup}\cdot$' and '${}^{\vee}\cdot$' the corresponding destructor functions. In particular, the application of the $L\Diamond_1$ rule to a chain type will trigger the application of the destructor function ${}^{\cup}\cdot$ to the chain.

We might think of the destructor ${}^{\cup}\cdot$ as implementing the partial deletion of information at each chain position (which can be thought of more intuitively as an implementation of Spell-Out). The corresponding constructor function would then implement the propagation of information along the chain, and the condition of Recoverability would reduce to something like the condition that the two functions cancel each other. As things stand, these functions are not well-defined, and any possible definition is likely to be rather context dependent. However, it does appear that the mathematical resources to implement spell-out and reconstruction are already built into the controlled multimodal approach to movement.

9 Missing Pieces, Future Work

¶16. So far we have skirted the issue of Head Movement. Chomsky (1995) makes fairly heavy use of head movement operations, but does little to advance the theory. In particular, it is not at all clear in a feature-driven theory of movement what features drive head movement. Stabler (forthcoming) makes appeal to the theory of Koopman (1994), who proposes a theory of head movement based on selection. This approach cause problems here, however. In a strictly resource conscious setting, a moving V would have to come out of the lexicon with two copies of the V feature, for example: one to allow the attachment of the VP in the higher phrase and one to allow the attachment of the V itself in the head of the higher phrase.

Lacking such a mystery feature and being unwilling to multiply category features, there is one possible solution left. Consider first our main verb—call it α—which is assigned type V/D. If it moves up to adjoin to a higher light verb, then it must be associated with a chain as its proof term. In particular, it is a function from D type arguments to a nontrivial chain.

$$(V/D)_i : \lambda x.\langle (t_\alpha > x), \alpha\rangle \tag{39}$$

The verb it combines with—call it β—must be such as to combine with a phrase and absorb its head.

$$(K\backslash(D\backslash V))/V : \lambda x_1.\lambda x_2.\lambda x_3.(x_3 \lhd (x_2 \lhd ((\pi_2 x_1 \blacktriangleleft \beta) > \pi_1 x_1))) \tag{40}$$

The appearance of the π functions is a symptom that our strict use of resources has been weakened here. It's just that we are weakening them in the proof term algebra.

Just for completeness, consider the possibility that the upper V must move (overtly) as well. Then it too must be a function to a nontrivial chain, whose first argument is still a phrase-moving head pair.

$$(K\backslash(D\backslash V))/V : \lambda x_1.\lambda x_2.\lambda x_3.\langle (x_3 \lhd (x_2 \lhd (t_\beta > \pi_1 x_1))), (\pi_2 x_1 \blacktriangleleft \beta)\rangle \tag{41}$$

Supposing that our inner V has combined with a DP γ, then applying this function to

$$\langle (t_\alpha > \gamma), \alpha\rangle \tag{42}$$

yields

$$\lambda x_2.\lambda x_3.\langle (x_3 \lhd (x_2 \lhd (t_\beta > (t_\alpha > \gamma)))), (\alpha \blacktriangleleft \beta)\rangle \tag{43}$$

This all works as far as I can tell, but leaves one feeling that the linguistics is now all being done in the term algebra and not in the deductive system.

¶17. We have also done nothing to implement successive cyclic movement. In Chomsky (1995) this is done by deleting (or erasing) only the feature on the target. In our terms this amounts to allowing moving features access to the structural rule of Contraction as well as movement.

This disturbs the rather neat relation between derivation and representation which can be observed here. In particular, we can no longer think of the lexical formula as determining the movement potential of the phrase it heads. In particular, since Contraction is a structural rule, it would not normally be taken to affect the proof terms of the structures it affects, but in our setting it is crucial to the derivation-representation connection that applications of Contraction create new chain positions in the proof term as well as new copies of the formula in the sequent.

One possible alternative would be to capitalize on the operation of Select, which is supposed to be the operation which introduces lexical item instances into the numeration, possibly selecting inflectional features at the same time. We might then appeal to Select to introduce lexical formulas with exactly as many copies of the crucial features as necessary. Computationally, however, this would introduce a very great deal of nondeterminism into the system.

¶18. Finally, we have done nothing to implement the distinction between overt and covert movement. In particular, we have not implemented the minimalist idea that features come in strong and weak varieties, and that only strong features force overt movement. Part of the reason for this is that the theory is somewhat vague on this point. In various places, Chomsky seems to want "strength" itself to be a feature. So it is not yet clear whether "strength" should be treated in a resource conscious manner, or rather in an additive manner, or by some other means completely (e.g., with yet more modal control).

References

Bobaljik, J. (1995). In terms of merge: Copy and head movement, *MIT Working Papers in Linguistics 27*, MIT Linguistics Dept., pp. 41–64.

Brody, M. (1995). *Lexico-Logical Form: A Radically Minimalist Theory*, MIT Press, Cambridge, MA.

Chomsky, N. (1995). *The Minimalist Program*, MIT Press, Cambridge, MA.

Collins, C. (1997). *Local Economy*, MIT Press, Cambridge, MA.

Koopman, H. (1994). Licensing heads, *in* D. Lightfoot and N. Hornstein (eds), *Verb Movement*, Cambridge University Press, Cambridge, pp. 261–296.

Moortgat, M. (1996). Categorial type logics, *in* J. van Benthem and A. ter Meulen (eds), *Handbook of Logic and Language*, Elsevier, chapter 2.

Moortgat, M. and Oehrle, R. T. (1997). Structural abstractions, *Proceedings of the Roma Workshop.*

Morrill, G. (1994). *Type Logical Grammar*, Kluwer, Dordrecht.

Stabler, E. P. (1996). Derivational minimalism, *Proceedings of LACL-96*, Nancy, France.

Stabler, E. P. (1997). Computing quantifier scope, *in* A. Szabolcsi (ed.), *Ways of Scope Taking*, Kluwer, Boston.

Stabler, E. P. (forthcoming). *Acquiring and Parsing Languages with Movement*, Basil Blackwell, Oxford.